TITLE OF THE INVENTION

Electronic Apparatus Having Game and Telephone Functions

Field of the invention

The illustrative embodiments relate to an electronic apparatus having a game function and a phone function. More specifically, the illustrative embodiments relate to an electronic apparatus pausing a game when receiving an incoming call.

Description of the prior art

An example of one such electronic apparatus is disclosed in a Japanese Patent Laying-open No.8-84172 published on March 26, 1996. The prior art is provided with a phone function and a radio receiving function, and when there is an incoming call while the phone is receiving the radio signal, the reception of the radio signal is ceased, and the phone processes the connection with the calling source in response to the pressing of a talk/hook switch. However, since the focus of the prior art is the cessation of the radio reception, when applying the prior art to an electric apparatus provided with a phone function and a game function, a game is ceased in response to the reception of an incoming call. Then the game has to be resumed from the beginning after completing the call.

BACKGROUND AND SUMMARY OF THE INVENTION

Therefore, it is a feature of the illustrative embodimentsto provide a novel electronic apparatus.

It is another feature of the illustrative embodiments to provide an electronic apparatus capable of restarting a game from the middle after the completion of a call.

According to an illustrative embodiment, an electronic apparatus having a game

function and a phone function comprises: an interruption key to instruct the game in progress to pause; a first register to which a first predetermined value is set in response to an operation of the interruption key; interrupting means for interrupting the game when the first register indicates the first predetermined value; detecting means for detecting an incoming phone call; and setting means for setting the first predetermined value into the first register in response to the detection of the incoming call.

When the interruption key to interrupt the game in progress is operated, the first predetermined value is set to the first register. Thereupon, the game is interrupted by the interrupting means. On the other hand, if an incoming phone call is detected by the detecting means, , the first predetermined value is set to the first register by the setting means when receiving the incoming call. Accordingly, when there is an incoming call, the game is interrupted without having to operate the interruption key.

In one aspect of an illustrative embodiment, when the first register indicates the first predetermined value, game data at this moment is stored in a memory by a storing means.

In another aspect of an illustrative embodiment, whether or not the game is in execution is determined by determination means. The setting means sets the first predetermined value in the first register during execution of the game. Specifically, when there is an incoming call during the execution of the game, the first predetermined value is set in the first register, whereby the game is interrupted.

In another aspect of an illustrative embodiment, when a restart key, to indicate a restart of the game, is operated, a second predetermined value is set to a second register. Thereupon, the game is restarted by the restart means.

In further aspect of an illustrative embodiment, when receiving an incoming call, a tone of the game screen is modified by a modifying means.

In another aspect of an illustrative embodiment, the game is executed by a game program downloaded by the phone function.

According to the illustrative embodiments, when there is an incoming call, the game is interrupted without operating the interruption key. More specifically, the game is not ceased but paused. Accordingly, the game can be restarted from the middle after completion of the call.

The first predetermined value is not only set in the register when the interruption key is operated, but also when there is an incoming call. Accordingly, even if a conventional game program (a program to interrupt a game when a first predetermined value is set in a register) is utilized without a modification, it is possible to interrupt the game upon receiving an incoming call. More specifically, the game can be paused when there is an incoming call without modifying a game pause function originally provided in the conventional game program.

According to an illustrative embodiment, a processing method of an electronic apparatus having a game function and a phone function, and provided with an interruption key, comprises the steps of: (a) interrupting a game in progress in response to an operation of the interruption key; (b) setting operation data of the interruption key when there is an incoming phone call. When the interruption key is operated, the game in progress is interrupted. Furthermore, when there is an incoming phone call, the operation data of the interruption key is set. Thus, when receiving the incoming phone call, processing is executed in a manner similar to that at the time the interruption key is operated.

In one aspect of the illustrative embodiments, the game data at this moment is stored in the memory in response to the operation of the interruption key or the incoming phone call.

In another aspect of the illustrative embodiments, the tone of a game screen in progress is modified in response to the incoming call of the phone.

Also, in an illustrative embodiment, when there is an incoming call, the game is interrupted without operating the interruption key. The game can be restarted from the middle of the game by operating the restart key upon completion of communication.

The above described objects and other objects, features, aspects and advantages of the illustrative embodiments will become more apparent from the following detailed description of the illustrative embodiments when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is an appearance view showing one embodiment of the illustrative embodiments;

Figure 2 is an illustrative view showing one example of an incoming call standby screen;

Figure 3 is an illustrative view showing one example of a menu screen;

Figure 4 is an illustrative view showing one example of a game screen;

Figure 5 is an illustrative view showing one example of an incoming call guide screen;

Figure 6 (A) is an illustrative view showing one example of a player name input screen when inputting a player's name;

Figure 6 (B) is an illustrative view showing one example of the player name input screen after having finished inputting the player's name;

Figure 7 is a block diagram showing a configuration of Figure 1 embodiment;

Figure 8 is a block diagram showing a configuration of a game CPU;

Figure 9 is a block diagram showing a configuration of a key controller and its periphery;

Figure 10 is a block diagram showing a configuration of an LCD controller and its periphery;

Figure 11 is a block diagram showing a configuration of an image composite circuit;

Figure 12 is a mapping state of a memory which a game CPU core can access;

Figure 13 is an illustrative view showing a configuration of a register 80;

Figure 14 is an illustrative view showing a configuration of LCD control data;

Figure 15 is an illustrative view showing a configuration of game key registers 1 to 3;

Figure 16 is an illustrative view explaining an area formed on an LCD screen;

Figure 17 is an illustrative view explaining a color displayed on the LCD screen;

Figure 18 is an illustrative view showing one example of binary image data drawn in a frame memory 86;

Figure 19 is an illustrative view showing one example of color image data drawn in a frame memory 88;

Figure 20 is an illustrative view showing another example of the binary image data drawn in the frame memory 86;

Figure 21 is an illustrative view showing another example of the color image data drawn in the frame memory 88;

Figure 22 is a flowchart showing a part of an operation of a phone CPU core;

Figure 23 is a flowchart showing another part of the operation of the phone CPU core;

Figure 24 is a flowchart showing the other part of the operation of the phone CPU

core;

Figure 25 is a flowchart showing further part of the operation of the phone CPU core;

Figure 26 is a flowchart showing another part of the operation of the phone CPU core;

Figure 27 is a flowchart showing the other part of the operation of the phone CPU core;

Figure 28 is a flowchart showing further part of the operation of the phone CPU core;

Figure 29 is a flowchart showing another part of the operation of the phone CPU core;

Figure 30 is a flowchart showing the other part of the operation of the phone CPU core;

Figure 31 is a flowchart showing further part of the operation of the phone CPU core;

Figure 32 is an illustrative view showing data set in a FIFO register 1 during key assignment processing;

Figure 33 is an illustrative view showing one example of a function assignment table;

Figure 34 is a flowchart showing another part of the operation of the phone CPU core;

Figure 35 is a flowchart showing the other part of the operation of the phone CPU core;

Figure 36 (A) is an illustrative view showing binary image data drawn in the frame memory 86 during player's name input processing;

Figure 36 (B) is an illustrative view showing color image data drawn in the frame memory 88 during the player's name input processing;

Figure 36 (C) is an illustrative view showing a player's name input screen displayed on the LCD;

Figure 37 (A) is an illustrative view showing binary image data drawn in the frame memory 86 during the player's name input processing;

Figure 37 (B) is an illustrative view showing color image data drawn in the frame memory 88 during the player's name input processing;

Figure 37 (C) is an illustrative view showing a player's name input screen displayed on the LCD;

Figure 38 (A) is an illustrative view showing binary image data drawn in the frame memory 86 during the player's name input processing;

Figure 38 (B) is an illustrative view showing color image data drawn in the frame memory 88 during the player's name input processing;

Figure 38 (C) is an illustrative view showing a player's name input screen displayed on the LCD;

Figure 39 is a flowchart showing a part of an operation of a game CPU core;

Figure 40 is a flowchart showing another part of the operation of the game CPU core;

Figure 41 is a flowchart showing the other part of the operation of the game CPU core;

Figure 42 is a flowchart showing further part of the operation of the game CPU core;

Figure 43 is a flowchart showing another part of the operation of the game CPU core;

Figure 44 is an illustrative view showing one part of an operation relating to another embodiment of the illustrative embodiments; and

Figure 45 is an illustrative view showing another part of the operation relating to another embodiment of the illustrative embodiments.

DETAILED DESCRIPTION OF THE DRAWINGS

Referring to Figure 1, a portable communication terminal with game function 10 of this embodiment includes a long thin rectangular formed main body 12. The main body 12 has an antenna 14 protruded from the upper surface thereof, and is provided with a speaker 16, a color LCD 18, a game start key 20, a game key 22, a phone key 24, and a microphone 26 on its front surface.

As the game key 22, there exists a cross key 22a which instructs a hero in a game to move in a predetermined direction (e.g. go ahead, go back, turnaround, jump and etc.) and an A button 22b and B button 22c which instruct the hero to act in a predetermined manner (e.g. obtainment of an item or attack a target). Furthermore, as the phone key 24, there exists a connect key 24a, a hold key 24b, number-keys 24c indicative of "0" to "9", a * key 24d, and a # key 24e.

An incoming call stand-by screen as shown in Figure 2 is displayed on the LCD 18 in an incoming call stand-by state. According to Figure 2, the day's date, current time, and a game character created by the applicant are displayed on the center of the screen, and characters indicative of a receiving state and a remaining amount of battery power are displayed on the upper portion of the screen. If an operator performs a dialing operation utilizing the connect key 24a and the number keys 24c in the incoming call stand-by state, the dialed telephone number is displayed on the LCD 18, and calling processing for an opponent is executed. If the opponent elects to receive the incoming call, a protocol is

established, and talking is enabled. On the other hand, if there is an incoming call from the opponent, an incoming call message and the telephone number of the transmission source are displayed on the LCD 18, and a ring tone is output from a speaker 48 (referring to Figure 7) provided on the back surface of the main body 12. If the operator herein operates the connect key 24a, the output of the ring tone is stopped, and the protocol is established between operator and the transmission source, leading to a communication enabled state.

If the operator operates the game start key 20 in the incoming call stand-by state, the operation mode is shifted from phone mode to game mode. First, a menu screen shown in Figure 3 is displayed. When the operator selects a desired game (e.g. Super Mario DX) from the menu screen, the selected game is started. The display on the LCD 18 is renewed to a game screen shown in Figure 4, and BGM is output from the speaker 48 on the back surface. The game proceeds in response to an operation of the game key 22 described above, and sound is output from the speaker 48 as necessary.

In this embodiment, in the game mode, the * key 24d functions as a start key, and the # key 24e functions as a select key (other phone keys may function as a start key and a select key). Accordingly, the game in progress is interrupted/resumed in response to an operation of the * key 24d. Specifically, when the * key 24d is pressed during the game, a pause state is set, whereby motion of the game screen is stopped, and the output of the BGM is interrupted. If the * key 24d is pressed again, the pause state is canceled, whereby the game screen starts to move, and output of the BGM is resumed. In addition, for example, a cursor on the screen is moved by operating the # key 24e on a player selection screen.

Even if the game is in progress, detection of an incoming call is executed, and when there is an incoming call, the motion of the game screen is stopped and the output of

the BGM is interrupted. That is, the game is paused in response to the incoming call. Furthermore, the ring tone is output from the speaker 48 in place of the BGM, and the display of the LCD 18 is renewed to an incoming call screen as shown in Figure 5. According to Figure 5, a tone of the game screen is changed, and an incoming call message and the telephone number of the transmission source are displayed on the game screen. Also, at this time, the output of the ring tone is interrupted by an operation of the connect key 24a, and the protocol is established, leading to a communication enabled state. After the communication is completed, the tone of the game screen is restored, and the game screen shown in Figure 4 is displayed again.

It is noted that the pause state is continued after the communication ends and the pause state is canceled in response to an operation of the * key 24d. More specifically, although the game is paused at the moment of the incoming call without operating the *key 24d, the pause state is canceled by operating the * key 24d. The game screen is moved and an output of the BGM is restarted as described above by canceling the pause state with the operation of the * key 24d.

If there is a need to register a player's name in the middle or at the start of the game, a player name input screen as shown in Figures 6 (A) or 6(B) is displayed on the LCD 18. At this time, the player's name is entered within a player's name input frame on the screen by operating the number keys 24c. Specifically, one or more alpha-numeric characters. is assigned to each number key 24c, and inputting a desired name is accomplished by operating the number keys 24c.

An inside of the main body 12 is specifically constituted as shown in Figure 7. The game start key 20 and the phone key 24 are directly connected to a CPU 28 for phone processing (phone CPU). Furthermore, the antenna 14 is connected to the phone CPU 28 via a transmitting / receiving circuit 32, and the speaker 16 and the microphone 26 are

connected to the phone CPU 28 via a sound codec 30. The speaker 48 is connected to the phone CPU 28 via an accumulator 46. An SRAM 34 and a flash memory 36 are connected to the phone CPU 28 via an 8-bit bus 38. The flash memory 36 is stored with a program for phone processing (phone program) and telephone directory data, and the SRAM 34 is utilized as a work RAM while processing the phone program.

On the other hand, a CPU 40 for game processing (game CPU) is directly connected with the game key 22, the LCD 18, an SRAM 42 and an external terminal S and is connected with the speaker 48 via the accumulator 46 and an amplifier 44. The SRAM 42 is utilized as an external work RAM when processing a game program.

Between the phone CPU 28 and the game CPU 40, a direct correspondence of an interruption request and transmission data is performed in addition to the correspondence of data via the 8bit-bus 38. The transmission data is also corresponded with an external apparatus (not shown) connected through the external connection-use terminal S.

The game CPU 40 is constituted as shown in Figure 8. A game CPU core 50, a sound processing unit (SPU) 64, a picture processing unit (PPU) 78, and an LCD controller 84 are included as processors. In addition, a system ROM 52, a flash memory 54, a work RAM 56, a color palette RAM 68, registers 70 and 80, VRAMs 72 and 74, an OAM (Object Attribute Memory) 76, frame memories 86 and 88, and a game key register 1 (referring to Figure 9) provided on a key controller 82 are included as external memories. Besides such processors and internal memories, a UART 58 and a UART selector 60 are provided for corresponding the above-described transmission data with the phone CPU 28 or the external apparatus, and an external SRAM I/F 62 for accessing the SRAM 42 shown in Figure 7 is also provided.

Memories which the game CPU core 50 can access are the system ROM 52, the flash memory 54, the work RAM 56, the SRAM 42, the color palette RAM 68, the

register 70, the VRAMs 72 and 74, the OAM 76, a game CPU control register 80d, a FIFO register 80e, an interruption register 80g provided in the register 80 (referring to Figure 13), and the game key register 1 provided in the key controller 82 (referring to Figure 9).

Memories to which the PPU 78 can gain access are the color palette RAM 68, the register 70, the VRAMs 72 and 74, and the OAM 76.

Memories to which the LCD controller 84 can gain access are the frame memories 86 and 88 and a frame memory register 80a, an output control register 80b, and a palette control register 80c provided in the register 80 (referring to Figure 13).

It is noted that although the SPU 64 can gain access to a sound ROM not shown, detailed description about sound processing will be omitted.

A memory mapping viewed from the game CPU core 50 is shown in Figure 12. According to Figure 12, the system ROM 52 is assigned to "00000000h"～"01999999h", the SRAM 42 is assigned to "02000000h"～"02999999h", and the work RAM 56 is assigned to "03000000h"～"03999999h". Various kinds of registers (70, 80d, 80e, 80g, and the game key register 1) are assigned to "04000000h"～"04999999h", and the color palette RAM 68 is assigned to "05000000h"～"05999999h". Furthermore, the VRAMs 72 and 74 are assigned to "06000000h"～"06999999h", the OAM 76 is assigned to "07000000h"～"07999999h", and the game program (image) is assigned to "08000000h"～"08199999h".

In addition, a browser program and a monitor program stored in the flash memory 54 are assigned to "10000000h"～"Xh-1", the game program stored in the flash memory 54 is assigned to "Xh" ～ "Xh+00199999h", and a backup stored in the flash memory 54 is assigned to "Xh+00200000h"～"10400000h". The game CPU core 50 gains access to respective memories utilizing such addresses.

In describing each memory in detail, the system ROM 52 stores a boot program of the game CPU core 50, a recovery program in the case of damage to the flash memory 54, and etc. Meanwhile, the flash memory 54 is, as described above, stored with the backup, the game program, the browser program, and the monitor program. The game program is downloaded from a server which is provided on the Internet and stored with a plurality of game programs. More specifically, if connecting to the server, utilizing the phone function so as to select a desired game program, the selected game program is downloaded to the flash memory 54 via the phone CPU 28. The work RAM 56 and the SRAM 42 are temporarily stored with data obtained during the processing of the above-described programs.

The VRAM 72 is stored with background image data indicative of characters in a static state (static characters), the VRAM 74 is stored with object image data indicative of characters with movements (dynamic characters), and the OAM (Object Attribute Memory) 76 is stored with an attribute of each dynamic character. For example, when the game screen shown in Figure 4 is displayed, the background image data stored in the VRAM 72 includes static character data such as a tree, a mountain, and earth. Meanwhile, the object image data stored in the VRAM 74 includes dynamic character data such as a hero jumping up in the air, a mushroom moving on the earth, and blocks floating in the air and moving slightly when pushed from underneath.

The color palette RAM 68 is stored with palette data for coloring the static character and the dynamic character, and the register 70 has a setting for designating an image mode and subjecting the screen to special effects.

Referring to Figure 13, the frame memory register 80a has address data of the frame memories 86 and 88 and image data. More specifically, an X address and a Y address of the frame memory 86 and one dot (=8 bit) of binary image data written into the

addresses (X, Y), and an X address and a Y address of the frame memory 88 and one dot (=8bit) of color image data written into the addresses (X, Y) are set. Setting to the frame memory register 80a is executed by the phone CPU 28. The address data and the image data for both of the frame memories 86 and 88 are set by the phone CPU 28 in the phone mode, and only the address data and the image data for the frame memory 86 are set by the phone CPU 28 in the game mode.

The output control register 80b is a register for controlling an output of image data from the LCD controller 84 to the LCD 18, and composition start line data, composition end line data, image start line data and LCD control data are set thereto. Referring to Figure 16 and Figure 17, an image based on the image data drawn in the frame memory 86 is displayed on an area 1 forming the display screen, and a composite image based on the respective image data drawn in the frame memories 86 and 88 is displayed on an area 2 forming the display screen. The composition start line data and the composition end line data indicate a start position and an end position of the area 2, respectively. Meanwhile, the image start line data is data to designate from which line of the frame memory 88 reading-out of the color image data is started. A scroll display of a color image is made possible by renewing the image start line data per a predetermined number of lines.

The LCD control data is specifically displayed by 8 bits as shown in Figure 14. The zero bit (=LCNT 0) is a mode selection flag, and "0" indicates the phone mode and "1" indicates the game mode. The first bit (=LCNT 1) is a flag to control an output / interruption of an image based on the binary image data drawn in the frame memory 86, and "0" indicates the interruption and "1" indicates the output. The second bit (=LCNT 2) is a flag to control whether or not blending is subjected to the color image when an image based on the color image data drawn in the frame memory 88 is displayed, and "0"

indicates a normal and "1" indicates the blending. The fourth bit to the third bit (=LCNT 4, 3) are a flag to control an output and a tone (color tone) of the image based on the color image data drawn in the frame memory 88, and "00" indicates an interruption of the output, "01" indicates a tone down, "10" indicates an equalized tone and "11" indicates a tone up. The fifth bit (=LCNT 5) is a flag to control a display on/off state of the LCD 18, and "0" indicates the display off state and "1" indicates the display on state. The sixth and seventh bits are not utilized in this illustrative embodiment.

The palette control register 80c is a register to designate a color of the image based on the binary image data drawn in the frame memory 86. Palette data 0 for specifying a background color and palette data 1 for specifying a character color are set with regard to the area 1 which is not a composition area. On the contrary, only the palette data 1 for specifying the character color is set with regard to the area 2 which is a composition area, and the background is always a transparent color.

The game CPU control register 80d has a game CPU start flag only. The game CPU start flag is set to "1" in response to an operation of the game start key 20, and set to "0" in response to an operation of the hold key 24b. The "1" indicates the game mode and the "0" indicates the phone mode.

The FIFO register 80e is a register to temporarily hold data corresponded between the game CPU 40 and the phone CPU 28 in the game mode. More specifically, it includes FIFO registers 1 and 2, each of which has a capacity of 64 $\times$ 16 bits. Since the phone CPU 28 of this embodiment is an 8 bit CPU, only 8 bits out of 16 bits are utilized. However, it is possible to respond to a phone CPU of 16 bits by setting the FIFO registers 1 and 2 to 16 bits.

The FIFO register 1 is used for data transmission from the game CPU 40 to the phone CPU 28, and the FIFO register 2 is used for data transmission from the phone CPU

28 to the game CPU 40. Accordingly, writing to the FIFO register 1 is executed only by the game CPU 40, and reading out from the FIFO register 1 is executed only by the phone CPU 28. Furthermore, writing to the FIFO register 2 is executed only by the phone CPU 28, and reading out from the FIFO register 2 is executed only by the game CPU 40. These FIFO registers 1 to 2 are utilized in assignment processing for assigning the start function and the select function to the * key 24d and the # key 24e, respectively, transmission processing of key information when the * key 24d or # key 24e is operated, clearing processing of the function assignment, and player name input processing.

The game key register 80f is a register to hold data in relation to the game operation, and includes game key registers 2 and 3. The game key register 1 is provided in the key controller 80 as described above, and all of the game registers 1 to 3 are formed by 8 bits as shown in Figure 15. The A button 22b is assigned to the zero bit (=K0, KI0, KO0), the B button 22c is assigned to the first bit (=K1, KI1, KO1), and the upper direction, the lower direction, the left direction and the right direction of the cross key 22a are assigned to the second bit to the fifth bit (K2 ~K5, K12~K15, KO2~KO5). The start key (* key 24d) is assigned to the sixth bit (=K6, KI6, KO6), and the select key (# key 24e) is assigned to the seventh bit.

The zero bit is renewed from "0" to "1" by operating the A button 22b, and the first bit is renewed from "0" to "1" by operating the B button 22c. The second bit is renewed from "0" to "1" when the cross key 22a is operated in the upper direction, and the third bit is renewed from "0" to "1"when the cross key 22a is operated in the lower direction. Furthermore, the fourth bit is renewed from "0" to "1"when the cross key 22a is operated in the left direction, and the fifth bit is renewed from "0" to "1"when the cross key 22a is operated in the right direction. In addition, the sixth bit is renewed from "0" to "1" when the * key 24d is operated, and the seventh bit is renewed from "0" to "1" when the # key

24e is pressed.

The interruption register 80g is a register to temporarily hold an interruption request transmitted between the game CPU 40 and the phone CPU 28. The interruption request generated from the game CPU 40 to the phone CPU 28 is set to an interruption register 1, and the interruption request from the phone CPU 28 to the game CPU 40 is set to an interruption register 2. It is noted that either the interruption generated from the phone CPU 28 to the game CPU 40 or the interruption from the game CPU 40 to the phone CPU 28 is utilized in the assignment processing to assign the start function and the select function to the * key 24d and the # key 24e, respectively, the transmission processing of key information when the * key 24d or the # key 24e is operated, the clearing processing of the function assignment, and the player's name input processing.

The key controller 82 and its periphery are constituted as shown in Figure 9. In the game mode, the * key 24d functions as a start key, and the # key 24e functions as a select key as described above. A predetermined bit of the phone key register 28b is renewed to "1" when the operator operates the * key 24d or the # key 24e. The phone CPU core 28a determines which key is operated from the setting of the phone key register 28b, and then renews the sixth bit of a game key register 3 to "1" when the * key 24d is operated, and renews the seventh bit of the game key register 3 to "1" when the # key 24e is operated.

The game key register 1 is set to a logical sum of a value corresponding to an operation of the game key 22 and a value output from the game key register 3. The game CPU core 50 inputs the setting of the game key register 1 and performs game processing according to the input setting. Accordingly, not only an operation of the game key 22, but also operations of the * key 24d and # key 24e are reflected on the game processing.

The setting of the game key register 1 is applied to the game key register 2, and therefore, the settings of the game key registers 1 and 2 always coincide with each other.

Accordingly, the setting of the game key register 2 is changed in response to the operation of the game key 22. The phone CPU core 28a determines whether the A button 22b is operated or not from the setting of the game key register 2 upon inputting a player's name, and executes processing according to the result of the determination.

Returning to Figure 8, the PPU 78 performs a process according to the palette data of the palette RAM 68 and the setting of the register 70 on the image data stored in the VRAMs 72 and 74, and outputs the 8 bits color image data thus processed to the LCD controller 84. However, it is only in the game mode that the background image data and the object image data are drawn in the VRAMs 72 and 74, and the PPU 78 transmits the color image data and the address data to the LCD controller 84 only in the game mode.

The LCD controller 84 is constituted as shown in Figure 10. Out of the LCD control data set in the output control register 80b, the LCNT0 is applied to a selector 84b, the LCNT 1 is applied to a reading controller 84f, the LCNTs 4 and 3 are applied to a tone control circuit 84g, and the LCNTs 2 and 5 are applied to an image composition circuit 84h. Furthermore, the composition start line data and the composition end line data set in the output control register 80b are applied to the image composition circuit 84h, and the image start line data is applied to a reading controller 84c.

In addition, the palette data 0 for the area 1, the palette data 1 for the area 1 and the palette data 1 for the area 2, set in the palette control register 80c, are applied to the composition image circuit 84h, and the X address for the frame memory 86, the Y address for the frame memory 86, the one dot of binary image data, X address for the frame memory 88, the Y address for the frame memory 88, and the one dot of color image data set in the frame memory register 80a are applied to a frame memory transmission controller 84a.

As described above, the binary image data and its address data, and the color

image data and its address data are set to the frame memory register 80a in the phone mode. At this time, the frame memory transmission controller 84a applies the binary image data and its address data to a writing controller 84e and applies the color image data and its address data to the selector 84b. The selector 84b selects the output of the frame memory transmission controller 84a when the LCNT 0 indicates "0" i.e. the phone mode. The color image data and the address data output from the frame memory transmission controller 84a are applied to the writing controller 84d through the selector 84b.

On the other hand, in the game mode, only the binary image data and its address data are set to the frame memory register 80a, and the frame memory transmission controller 84a applies the binary image data and the address data to the writing controller 84e. The color image data and the address data are applied from the PPU 78 to the selector 84b. The LCNT 0 of the LCD control data indicates "1" in the game mode, and the selector 84b selects an output from the PPU 78 at this time. Accordingly, the output from the PPU 78 is applied to the writing controller 84d via the selector 84b in the game mode.

As described later in detail, the address data indicates only a writing start address with regard to either the binary image data or color image data. Accordingly, either the writing controller 84e or 84d renews the writing address at every dot in reference to the applied address data, and writes the one dot of binary image data or the one dot of color image data to the renewed address which is sequentially applied. Thus, the binary image data and the color image data of a predetermined number of dots are stored in the frame memories 86 and 88, respectively.

It is noted that although either the frame memories 86 or 88 has a capacity capable of storing 160 $\times$ 160 dots of image data, 1 bit is assigned to each dot of the frame memory

86, and 8 bits are assigned to each dot of the frame memory 88. Accordingly, each address of the frame memory 86 is written with the one dot of binary image data, and each address of the frame memory 88 is written with the one dot of color image data.

The reading controller 84f reads the binary image data from the frame memory 86 and applies the read binary image data to the image composition circuit 84h when the LCNT 1 of the LCD control data indicates "1". On the other hand, the reading of the binary image data from the frame memory 86 is suspended when the LCNT 1 indicates "0". The reading controller 84c specifies a reading start line of the frame memory 88 by the image start line data and starts to read out the color image data from the specified reading start line. The read color image data is applied to the image composition circuit 84h through the tone control circuit 84g.

The tone control circuit 84g executes a tone control or an output control in response to the values of the LCNTs 4 and 3 as follows. Specifically, if the LCNTs 4 and 3 indicates "01", an RGB level of the color image data is decreased (tone down), and the color image data of which tone is down is applied to the image composition circuit 84h. Furthermore, if the LCNTs 4 and 3 indicates "11", the RGB level of the color image data is increased (tone up), and the color image data of which tone is up is applied to the image composition circuit 84h. In addition, if the LCNTs 4 and 3 indicates "10", the color image data is applied to the image composition circuit 84h as it is, and if the LCNTs 4 and 3 indicates "00", the output of the color image data to the image composition circuit 84h is suspended.

The image composition circuit 84h is constituted as shown in Figure 11. A line counter 8402h counts the number of lines forming one frame and applies the counted value to a comparator 8403h. The comparator 8403h is also supplied with the composition start line data and the composition end line data set in the output control

register 80b. The comparator 8403h changes the output to the high level when the counted value of the line counter 8402h coincides with a data value of the composition start line data, and restores the output from the high level to the low level when the counted value of the line counter 8402h coincides with the data value of the composition end line data+1. Accordingly, the output of the comparator 8403h becomes low in level from the start position to the end position of the area 1 shown in Figure 16 and becomes high in level from the start position to the end position of the area 2 shown in Figure 16.

A selector 8401h selects the palette data 0 for the area 1 set in the palette control register 80c when the output of the comparator 8403h is low in level, and selects the color image data output from the tone control circuit 84g when the output of the comparator 8403h is high in level. The output of the selector 8401h is applied to a selector 8407h.

A blending circuit 8404h performs a blending process on the palette data 1 for the area 2 set in the palette control register 80c and the color image data output from the tone control circuit 84g. A selector 8405h selects the palette data 1 for the area 2 when the LCNT 2 of the LCD control data indicates "0", and selects the output of the blending circuit 8404h when the LCNT 2 indicates "1". The output of the selector 8405h is applied to a selector 8406h. The output of the comparator 8403h is also applied to the selector 8406h. The selector 8406h selects the palette data 1 for the area 1 set in the palette control register 80c when the output of the comparator 8403h is low in level, and selects the output of the selector 8405h when the output of the comparator 8403h is high in level.

The selector 8407h receives the binary image data read out by the reading controller 84f, and selects anyone of the selectors 8401h and 8406h according to a data value of each dot forming the binary image data (bit value). More specifically, the output of the selector 8401h is selected when the bit value indicates "0", and the output of the selector 8406h is selected when the bit value indicates "1".

A switch 8408h is in an off state when the LCNT 5 of the LCD control data indicates "0" and is in an on state when the LCNT 5 indicates "1". Accordingly, only when the LCNT 5 indicates "1", the output of the selector 8407h is applied to the LCD 18 via the switch 8408h.

In the area 1, the selector 8401h selects the palette data 0 for the area 1, and the selector 8406h selects the palette data 1 for the area 1. The selector 8407h selects the palette data 0 for the area 1 when each bit value of the binary image data is "0" and selects the palette data 1 for the area 1 when each bit value indicates "1". As a result, the characters indicative of the receiving condition and the remaining amount of the battery are displayed on the area 1 as shown in Figure 17.

In the area 2, the selector 8401h selects the color image data, and the selector 8406h selects the output of the selector 8405h. When not performing the blending process, the palette data 1 for the area 2 is applied from the selector 8405h to the selector 8406h. The selector 8407h selects the color image data when each bit value of the binary image data is "0"and selects the palette data 1 for the area 2 when each bit value indicates "1". As a result, the characters indicative of the date and the current time and the color image are displayed on the area 2 as shown in Figure 17.

The incoming call stand-by screen shown in Figure 2 is formed by the binary image data indicative of the receiving state, the remaining amount of the battery, the date, the current time shown in Figure 18 and the color image data indicative of the game character shown in Figure 19. The phone CPU 28 shown in Figure 7 sets the address data corresponding to a head of the area 1 into the frame memory register 80a, and sets the binary image data indicative of the date and the current time (display at the area 1) into the frame memory register 80a dot by dot. Successively, the phone CPU 28 sets the address data corresponding to a head of the area 2 into the frame memory register 80a and

sets the binary image data indicative of the date and the current time (display at the area 2) into the frame memory register 80a dot by dot. The LCD controller 84 repeatedly reads the binary image data renewed dot by dot from the frame memory register 80a and writes the read binary image data of each dot to the frame memory 86 based on the address data. Thus, the binary image data shown in Figure 18 is obtained in the frame memory 86.

The phone CPU 28 sets the address data corresponding to the head of the area 2 and the color image data of each dot indicative of the game character into the frame memory register 80a following the setting of the binary image data. The LCD controller 84 repeatedly reads the color image data renewed dot by dot from the frame memory register 80a and writes the read color image data to the frame memory 88 based on the address data. Thus, the color image data shown in Figure 19 is obtained in the frame memory 88.

The LCD controller 84 , based on the settings of the output control register 80b and the palette control register 80c, reads the binary image data and the color image data from the frame memories 86 and 88, a tone control of the color image data, a coloring based on the binary image data, a composition of the binary image data and the color image data, and an output of the composite image data. As a result, the stand-by screen shown in Figure 2 is displayed on the LCD 18.

If the operator performs a dialing operation, the phone CPU 28 sets the address data corresponding to the head of the area 2 and the binary image data indicative of the telephone number of the opponents into the frame memory register 80a. The LCD controller 84 writes the binary image data to the frame memory 86 in the same manner as the above description. Accordingly, the binary image data indicative of the date and the current time is renewed by the binary image data indicative of the telephone number of the opponent. As a result, the date and the current time displayed on the LCD 18 is also

switched to the telephone number of the opponent.

The phone CPU 28 also establishes a connection with the opponent through the transmitting receiving circuit 32 and the antenna 14. When the connection is established, sound data of the opponent is received, and the received sound data is output from the speaker 16 via the sound codec 30. Furthermore, sound data of the operator, input through the microphone 26 is output from the antenna 14 via the sound codec 30 and the transmitting and receiving circuit 32.

On the other hand, if there is an incoming call through the antenna 14 and the transmitting receiving circuit 32, the phone CPU 28 sets the address data corresponding to the head of the area 2 and the binary image data indicative of the incoming call message and the telephone number of the transmission source into the frame register 80a. The LCD controller 84 writes the binary image data to the frame memory 86, whereby the binary image data indicative of the date and the current time is renewed by the binary image data indicative of the incoming call message and the telephone number of the transmission source. Thus, the date and the current time displayed on the LCD 18 are also renewed to include the incoming call message and the telephone number of the transmission source. Furthermore, the phone CPU 28 applies ring tone data to the speaker 48 through the adder 46. The ring tone is output from the speaker 48.

If the operator herein performs a receiving operation, the phone CPU 28 stops the output of the ring tone data, and establishes the connection with the transmission source. After the connection is established in the same manner as the above description, the sound data of the opponent is output from the speaker 16 via the sound codec 30, and the sound data of the operator input through the microphone 26 is output from the antenna 12 through the sound codec 30 and the transmitting and receiving circuit 32.

If the operator operates the game start key 20 in the incoming call stand-by state,

the phone CPU 28 sets the game start flag. Furthermore, the phone CPU 28 sets the address data corresponding to the head of the area 2 and the binary image data in which all dots indicate "0" into the frame memory register 80a and sets "1" into the LCNT 0 of the LCD control data. The LCD controller 84 writes the binary image data in which all dots indicate "0" into the frame memory 86 based on the address data, and selects the PPU 78 as an input destination of the color image data.

On the other hand, the game CPU core 50 draws the background image data and the object image data forming the game screen (color image) in the VRAMs 72 and 74 respectively, and sets the palette data of the static character and the dynamic character into the color palette RAM 68. Furthermore, the attribute of the dynamic character is set into the OAM 76 and the settings with respect to the output of the background and the object are set into the register 70. As a result, the color image data shown in Figure 21, for example, is formed by the PPU 78, and output to the LCD controller 84. The LCD controller 84 writes the color image data output from the PPU 78 to the frame memory 88. The frame memories 86 and 88 are stored with the binary image data shown in Figure 20 and the color image data shown in Figure 21, respectively.

The LCD controller 84, in such a manner as described above, reads the binary image data and the color image data from the frame memories 86 and 88, performs the tone control according to the values of the LCNTs 4 and 3 of the LCD control data, and composites the palette data with the color image data on the basis of the binary image data, the composition start line data and the composition end line data. As a result, the game screen shown in Figure 4 is displayed on the LCD 18.

The game CPU core 50 also applies a key code indicative of the * key 24d and the # key 24e and function information assigned to the * key 24d and the # key 24e to the phone CPU 28 through the FIFO register 1 shown in Figure 9. Herein, the function

information assigned to the * key 24d indicates "START", and the function information assigned to the # key 24e indicates "SELECT". Accordingly, the phone CPU 28 sets "1" into the start key bit KO6 of the game key register 3 when the * key 24d is operated and sets "1" into the select key bit KO7 of the game key register 3 when the # key 24e is operated. Each bit of the game key register 1 is set with logical sum of the value corresponding to the operation of the game key 22 and the value output from the game key register 3. The game CPU core 50 performs the game processing according to the setting of the game key register 1.

After the game is started, in response to an operation of the game key 22, the background image data, the object image data, and the attribute of the object are renewed, and furthermore, the settings of the color palette register 68 and the register 70 are renewed. As a result, the color image data output from the picture processing unit 78 is also changed, and movements are generated on the game screen. On the other hand, the effect sound data is output from the SPU 64 as necessary, and the effect sound is generated from the speaker 48.

If the * key 24d is operated during the progress of the game, the game CPU 50 causes the SPU 64 and the PPU 78 to stop outputting the BGM data and the color image data so as to shift to the pause state. Therefore, the output of the BGM and the movements of the game screen are stopped. If the * key 24d is operated again, the game CPU 50 makes the SPU 64 and the PPU 78 resume the output of the BGM data and the color image data so as to cancel the pause state. The BGM is output from the speaker 48 again, and the game screen on the LCD 18 also starts to move again.

It is noted that if the menu screen is displayed during the progress of the game, the game CPU core 50 changes the setting of the register 70 in response to an operation of the #key 24e. The setting to be changed is a setting relating to the cursor displayed on the

menu screen, and the display position of the cursor is changed due to the change.

If there is the incoming call during the game, the phone CPU 28 sets "1" into the start key bit KO6 of the game key register 3 in addition to the above-described processing, and sets "11" into the LCNTs 4 and 3 of the LCD control data. Furthermore, the phone CPU 28 sets the binary image data indicative of the incoming call message and the telephone number of the transmission source and the address data corresponding to the head of the area 2 into the frame memory register 80a and applies the ring tone data to the speaker 48 via the adder 46. Additionally, the game CPU core 50 makes the SPU 64 stop outputting the BGM to the SPU 64 and stops its own process so as to pause the game.

The LCD controller 84 stores the binary image data set in the frame memory register 80a to the frame memory 86. Furthermore, the LCD controller 84 respectively reads the binary image data and the color image data from the frame memories 86 and 88, performs the torn up processing on the color image data, and composites the palette data with the color image data on the basis of the binary image data, the composition start line data and the composition end line data. As a result, the incoming call guide screen shown in Figure 5 is displayed on the LCD 18, and the output of the speaker 48 is switched from the BGM to the ring tone.

If the operator performs the receiving operation, the phone CPU 28 stops the output of the ring tone data and establishes the connection with the transmission source as described above. After the connection is established, the sound data of the opponent is output from the speaker 16 through the sound codec 30, and the sound data of the operator input through the microphone 26 is output from the antenna 14 via the sound codec 30 and the transmitting receiving circuit 32.

After finishing a communication, the phone CPU 28 restores the LCNTs 4 and 3 of the LCD control data to "10", and sets the address data corresponding to the head of the

area 2 and the binary image data in which all dots indicate "0" into the frame memory register 80a. As a result, the display of the LCD 18 returns to the game screen shown in Figure 4. It is noted the movements of the game screen have been stopped, and furthermore, the output of the BGM has been stopped. That is, the game retains the pause state. The pause state is canceled in response to an operation of the * key 24d by the operator.

More specifically, the phone CPU core 28a performs processing according to flowcharts shown in Figures 22 to 31 and Figures 34 and 35. First, the LCD control data set in the LCD control register 80c is set to an initial value "110010" in a step S1. That is, the LCD display on/off flag shown in Figure 14 is set to "1", the color image tone control flag is set to "10", the color image blending control flag is set to "0", the binary image output control flag is set to "1", and the mode selection flag is set to "0".

In a following step S2, subroutines shown in Figures 25 and 26 are processed so as to display the incoming call stand-by screen shown in Figure 2. First, the LCNT 5 of the LCD control data is renewed to "0" in a step S51 shown in Figure 25, whereby the display of the LCD 18 is in an off state. Then, a counted value N is set to "0" in a step S52, and the X address for the frame memory 86, the Y address for the frame memory 86 and the one dot of binary image data are set in the frame memory register 80a, respectively in steps S53, 54, and 55. More specifically, the head address of an area N formed in the frame memory 86 (corresponding to the area N on the display screen) and the one dot of binary image data to be written to the head address are set to the frame memory register 80a. In a following step S56, it is determined whether or not the writing of the binary image data to the area N is completed . If "NO", the process of the step S55 is repeated, and if "YES", the process proceeds to a step S57.

The LCD controller 84 first reads the head address data and the binary image data

set in the frame memory register 80a, and writes the binary image data to the address indicated by the head address on the frame memory 86. In completion of the writing one dot of binary image data, the LCD controller 84 renews the address of a writing destination by itself. Accordingly, there is no need to renew the writing address on the phone CPU 28 side, and the binary image data is written to desired addresses in the frame memory 86 by merely repeating the setting process of the binary image date in the step S55.

It is determined whether the counted value N reaches "2" or not in the step S57, and if "NO", the counted value N is incremented in a step S58, and then the processing of the steps 53 to 56 is repeated. Thus, the binary image data, indicative of the characters of the receiving state and the remaining amount of the battery, is drawn in the area 1 of the frame memory 86, and the binary image data indicative of the date and the current time is drawn in the area 2 of the frame memory 86. It is noted that although the image data of the whole screen is rewritten in this embodiment, only the area of image data required to change may be rewritten. (The same is applied to steps S59 to S62 relating to the area 2 described later.)

If "YES" is determined in the step S57, the X address for the frame memory 88, the Y address for the frame memory 88 and the one dot of color image data are set to the frame memory register 80a, respectively in the steps S59, S60 and S61. The X address and the Y address set in the steps S59 and S60 indicate the head address in the frame memory 88, and the one dot of color image data set in the step S61 is image data to be written to the head address. It is determined whether or not the writing of the color image data to the area 2 is completed in the step S62, and if "NO", the process of the step S61 is repeated. Therefore, the color image data of the game character shown in Figure 2 is drawn in the frame memory 88.

If "YES" is determined in the step S62, the palette data 0 for the area 1, the palette data 1 for the area 1, and the palette data 1 for the area 2 are set in the palette control register 80c in steps S63, S64, and S65, respectively. Whereby, the background color and the text color (character color) of the binary image displayed on the area 1 of the LCD18 and the text color of the binary image displayed on the area 2 are determined. The background color of the binary image on the area 2 is transparent as described above.

The composition start line data, the composition end line data, and the image start line data are set in the output control register 80c in steps S66, S67, and S68, respectively. The composition start line data indicates from which line of the binary image the composition of the color image is started, the composition end line data indicates at which line of the binary image the composition of the color image is completed, and the image start line data indicates from which line of the frame memory 88 the reading is started.

In completion of the process of the step S68, the process proceeds to a step S69, and the LCNT5 of the LCD control data is set to "1" so as to turn the display of the LCD 18 on. Therefore, the incoming call stand-by screen shown in Figure 2 is displayed on the LCD 18.

Returning to Figure 22, it is determined whether or not the game start key 20 is being operated, whether or not the hold key 24b is being operated, whether or not a dialing operation is being performed, and whether or not an incoming phone call occurs in steps S3, S7, S11 and S16, respectively. If the game start key 20 is being operated, "YES" is determined in the step S3, and the game CPU start flag shown in Figure 13 is set in a step S4, and then the LCNT 0 of the LCD control data is set to "1" in a step S5. Therefore, the game mode is started. When the process of the step S5 is completed, a subroutine shown in Figure 27 is processed in a step S6.

First, the LCNT5 of the LCD control data is renewed to "0" in a step S71 to turn

the display of the LCD 18 off. Next, the X address for the frame memory 86 and the Y address for the frame memory 86 are set to the frame memory register 80a in steps S72 and S73. The address set herein indicates the head address of the area 2 formed. The one dot of binary image data is set to the same frame memory register 80a in a step S74, and it is determined whether or not the writing of the binary image data to the area 2 is completed in a following step S75. If "NO", the process of the step S74 is repeated, and if "YES", the palette data 1 for the area 2 is set to the palette control register 80c in a step S76, and the LCNT5 of the LCD control data is restored to "1" in a step S77 to turn the display of the LCD 18 off. The processing of the subroutine of the step S6 enables the writing of the binary image data in which all dots indicate "0" to the area 2 of the frame memory 86. As a result, the date and the current time displayed on the incoming call stand-by screen is erased.

If the hold key 24b is operated, "YES" is determined in the step S7, and the game CPU start flag is reset in a step S8, and then, "0" is set to the LCNT0 of the LCD control data in a step S9. Thus the phone mode is started. In completion of the processing in the step S9, subroutines shown in Figures 28 and 29 are processed in a step S10 to display the incoming call stand-by screen on the LCD 18.

First, the LCNT5 of the LCD control data is renewed to "0" in a step S81 shown in Figure 28 to turn a display of the LCD 18 off. Next, the X address for the frame memory 86, the Y address for the frame memory 86 and the one dot of binary image data are set to the frame memory register 80a in steps S82, S83, and S84, respectively. The address data set in the steps S82 and S83 is the head address of the area 2 formed in the frame memory 86, and the binary image data set in the step S84 is data to be written to the head address. In a step S85, it is determined whether or not the writing of the binary image data to the area 2 is completed , and the process of the step S84 is repeated until determination of

"YES" is obtained. Thus, the binary image data of the date and the current time forming the incoming call stand-by screen are written to the area 2.

The X address for the frame memory 88, the Y address for the frame memory 88 and the one dot of color image data are set to the frame memory register 80a in following steps S86, S87 and S88. The X address and the Y address respectively set in the steps S86 and 87 indicate the head address of the area 2 formed in the frame memory 88, and the one dot of color image data set in the step S88 is image data to be written to the head address. In a step S89, it is determined whether or not the writing of the color image data to the area 2 is completed , and if "NO" is determined, the process of the step S88 is repeated. Thus, the color image data of the game character forming the incoming call stand-by screen is drawn in the frame memory 88.

If "YES" is determined in the step S89, the palette data 1 for the area 2 is set to the palette control register 80c in a step S90, and the composition start line data, the composition end line data, and the image start line data are set to the output control register 80b in steps S91, S92 and S93, respectively. The LCNT5 of the LCD control data is restored to "1" in a step S94. Thus the incoming call stand-by screen shown in Figure 2 is displayed on the LCD 18.

If the dialing operation is executed, "YES" is determined in the step S11, and the subroutine shown in Figure 27 is processed in a step S12. It is noted that the binary image data drawn in the frame memory 86 at this time is image data indicative of the phone number of the opponent. Accordingly, the phone number of the opponent is displayed on the LCD 18 instead of the date and the current time. When the processing in the step 12 is completed, communication processing is executed in a step 13, and it is determined whether or not the hold key 24b is operated in a step 14. Then, as long as the hold key 24b is not operated, the processing in the step 13 is repeated, and when the hold key 24b is

operated, the subroutine shown in Figure 27 is processed again in a step S15. At this time, the binary image data indicative of the date and the current time is drawn in the frame memory 86, and therefore, the display of the LCD 18 is renewed to the date and the current time from the phone number of the opponent.

If it is determined the incoming call of the phone occurs in a step S16 shown in Figure 23, a state of the game CPU start flag is determined in a step S17. If the game CPU start flag is in a reset state, the process directly proceeds to a step S20. On the other hand, if the game CPU start flag is in a set state, the start key bit KO6 of the game key register 3 shown in Figure 15 is set to "0" in a step S18, the LCNTs 4 and 3 of the LCD control data shown in Figure 14 are set to "11" in a step S19, and then the process proceeds to the step S20.

By the process in the step S18, the start key bit KO6 is set irrespective of an operation of the * key 24d (function as a start key), and the game proceeds to a pause state. Furthermore, the tone of the game screen is up by the processing in the step S19.

The subroutine shown in Figure 27 is processed in the step S20. At this time, the binary image data indicative of the incoming call message "IN INCOMING CALL" and the phone number of the transmission source is drawn in the frame memory 86. Successively, the ring tone is generated from the speaker 48 in a step S21, and whether the connect key 24a is operated or not is determined in a step S22. Then, where the connect key 24a is operated, the output of the ring tone is stopped in a step S23, and the communication processing is executed in a step S24. Whether the hold key 24b is operated or not is determined in a step S25, and as long as the hold key 24b is not operated, the communication processing in the step S23 is repeated.

If the hold key 24b is operated, the process proceeds from the step S25 to a step S26, and a state of the game CPU start flag is determined. If the game start flag is in the

reset state, the process proceeds to a step S28. However, if the game start flag is in the set state, the LCNT 4s and 3 of the LCD control data are set to "10" in a step S27, i.e. the tone of the game screen is restored, and then the process proceeds to the step S28. In the step S28, the subroutine shown in Figure 27 is processed, whereby the binary image data in which all dots indicate "0" is written to the frame memory 86. The incoming call message and the phone number of the transmission source displayed on the LCD 18 are deleted by the processing of the step S28. In completion of the processing in the step S28, the process proceeds to a step S33.

When "NO" is determined in any steps S3, S7, S11 and S16, the remaining amount of the battery and the receiving condition are detected in steps S29 and S30, and the today's date and the current time are detected in a step S31. Successively, a subroutine shown in Figure 30 is processed in a step S32, and in completion of the process of the step S32, the process proceeds to the step S33.

Referring to Figure 30, first the counted value N is set to "1" in a step S101, and then, the X address for the frame memory 86 and the Y address for the frame memory 86 are set in the frame memory register 80a in steps S102 and S103, respectively. Furthermore, the one dot of binary image data is set in the same frame memory register 80a in a step S104. The address data set in the steps S102 and S103 is the head address of the area N formed in the frame memory 86, and the binary image data set in the step S104 is data to be written into the head address. It is determined whether or not the writing of the binary image data to the area N is completed in a step S105, and if "NO", the process of the step S104 is repeated. If "YES" in the step S105, it is determined whether the counted value N is "2" or not in a step S106, and if "NO", the counted value N is incremented in a step S107, and then, the process returns to the step S102.

As a result, the processing from the steps S102 to S105 is repeated twice, and the

binary image data is written to the respective areas 1 and 2 formed in the frame memory 86. The area 1 is stored with the binary image data indicative of the characters of the receiving condition and the remaining amount of the battery, and the area 2 is stored with the binary image data indicative of the characters of the current date and time. The receiving condition, the remaining amount of the battery, the date and the time detected in the steps S29 to S31 are reflected on the respective characters.

If "YES" is determined in the step S106, the palette data 0 for the area 1, the palette data 1 for the area 1 and the palette data 1 for the area 2 are set to the palette control register 80c in steps S108, S109 and S110, respectively. The binary image data formed in the frame memory 86 is output to the LCD 18 through the coloring processing according to the palette data 0 for the area 1, the palette data for the area 1 and the palette data 1 for the area 2. The area 1 of the LCD 18 is displayed with the receiving condition and the remaining amount of the battery at this moment, and the area 2 is displayed with the characters indicative of the current date and time.

Referring to Figure 24, a state of the game CPU start flag is determined in the step S33, and if the flag is in the reset state, the process returns to the step S3. On the other hand, if the game CPU start flag is in the set state, key transfer processing (described after) is executed in a step S34, and it is determined whether or not the interruption request is set to the interruption register 1 in a step S35. If the interruption request is not set, the process directly returns to the step S3. Accordingly, when the phone mode is set, the processing from the steps S3 to S32 are repeated, and when the phone mode is set but the interruption request is not given, the processing from the steps S3 to S32 and the step S34 are repeated.

If the interruption request is set into the interruption register 1, the setting of the FIFO register 1 is detected in a step S36, and the contents of the detected setting are

identified in steps S37, S39, and S41.

If the content of the detected setting is a key assignment request, key assignment processing is executed in a step S38. Specifically, a subroutine shown in Figure 31 is executed. First, the setting is detected from the FIFO register 1 in a step S111, and it is determined whether the detected setting is an end code or not in a step S112. Referring to Figure 32, in a case the key assignment request is set in the FIFO register 1, a key code indicative of any one of the phone keys 24 is set following the key assignment request, and the function information indicative of a function desired to be assigned to the key code is set following the key code. An end code is set after repeating of such the key code and the function information.

Accordingly, the key code is detected in the step S111, and "NO" is determined in the first step S112. In a step S113, a key item corresponding to the setting detected i.e., the key code is specified from a function assignment table 34a shown in Figure33 (formed in the SRAM34). For example, the "# key" is specified where the lower four digits of the key code is "1010", and the "* key" is specified where the lower four digits of the key code is "1011". The setting of the FIFO register 1 is detected again in a step S114. As described above, the setting following the key code is the function information, and the function information is detected in the step S114. In a step S115, the function information detected is assigned to the key item specified in the step S113. In completion of the processing of the step S115, "0" is set to the corresponding bit of the game key register 3, and thereafter, the process returns to the S111. When the setting detected in the step S111 indicates the end code, "YES" is determined in the step 112, restoring to the hierarchical upper level of the routine.

For example, if the function information indicative of "select" (lower 8 bits are 10000000) is detected following the key code indicative of the "# key", the selection

function is assigned to the item of "# key" in the function assignment table 34a, and "0" is set in the KO7 of the game key register 3 shown in Figure 15. Meanwhile, if the function information indicative of "start" (lower 8bits are 01000000) is detected following the key code indicative of "* key", the start function is assigned to the item of "* key" in the function assignment table 34a, and "0" is set to the KO6 of the game key register 3.

A subroutine shown in Figure 34 is executed with reference to the function assignment table 34 thus formed in the step S34 shown in Figure 24. First, in a step S121, it is determined which phone key 24 is operated referring to the phone key register 28b shown in Figure 9. The function information assigned to the identified operation key is searched from the function assignment table 34a in a step S122, and it is determined whether or not the function information is found from the function assignment table 34a in a step S123.

According to the Figure 33, the key item indicative of the number keys 24c is provided to the function assignment table 34a besides "# key" and "* key", and some key items may be not assigned the function information. This is the reason why it is determined whether the function information is found or not in the step S123. Where the function information is not found, the process directly restores to the hierarchical upper level of the routine, and where the function information is found, the corresponding bit of the game key register 3 is set to "1" in a step S124, restoring to the hierarchical upper level of the routine.

For example, when the operator presses the number keys 24c indicative of "1", no bits of the game key register 3 are set to "1". On the other hand, when the operator presses the * key 24d, the KO6 of the game key register 3 is set to "1".

Returning to Figure 24, when the contents of the detected setting are a key assignment clearing, "YES" is determined in a step S39, and the function assignment

table 34a is cleared in a step S40. When the contents of the detected setting are a FEP (Front End Processing) transfer request, "YES" is determined in a step S41, and FEP transfer processing is executed in a step S42.

The FEP transfer processing is executed following a subroutine shown in Figure 35. First, the cursor display position data is detected from the FIFO register 1 in a step S131, and then, the subroutine shown in Figure 27 is processed in a step S132. The binary image data drawn in the area 2 of the frame memory 86 at this time is image data indicative of the cursor. A drawing position of the cursor is determined based on the cursor display position data, and the binary image data is stored in the frame memory 86 in a manner shown in Figure 36 (A).

On the other hand, color image data indicative of the guide message "INPUT YOUR NAME" and the name input frame are drawn in the frame memory 88 by the game CPU 40. As a result, the player name input screen shown in Figure 36 (C) is displayed on the LCD 18.

In completion of the process in the step S132, the settings of the phone key register 28b and the game key register 2 are detected in a step S133, and it is determined whether or not the A button 22b is operated in a following step S134. If "NO" is determined herein, it is assumed that the phone key 24 is operated, and then the process proceeds to a step S135 so as to execute the FEP processing. In completion of the FEP processing, the process returns to the step S132. A character string in correspondence to the operation of the phone key 24 is generated by the FEP processing, and the generated character string is displayed on the LCD 18 by processing of the following step S132.

For example, when inputting "YAMADATARO" by operating the phone key 24, the binary image data is drawn on the frame memory 86 in a manner shown in Figure 37 (A), and the characters of "YAMADATARO" are displayed at the name input frame on

the LCD 18 as shown in Figure 37(C). The cursor is displayed at the right of "RO".

If the operator operates the A button 22b after the completion of the name input, KI0 bit of the game key register 2 shown in Figure 15 is renewed to "1". At this time, "YES" is determined in the step S134, and "FEP result transmission" and a JIS code in correspondence to the character string generated are set into the FIFO register 2. Furthermore, the interruption request is set to the interruption register 2 in a step S138. A subroutine shown in Figure 27 is processed in a step S139, and thereafter, the process is restored to the hierarchical upper level of the routine.

The binary image data in which all bits indicate "0" is drawn in the area 2 of the frame memory 86 in the step S139, whereby only the binary image data of characters indicative of the receiving condition and the remaining amount of the battery remains in the frame memory 86. However, color image data indicative of "YAMADA TAROU" is drawn in the frame memory 88 in a manner shown in Figure 38 (B) by a process of the game CPU 40 described later, and the player name input screen shown in Figure 38 (C) is displayed on the LCD18.

Returning to Figure 24, in completion of the processing in the step S38, S40, or S42, the interruption register 1 is reset in a step S44. Meanwhile, if "NO" is determined in the steps S37, S39, and S41, the corresponding process is executed in a step S43, and the interruption register 1 is reset in the step S44. In completion of the processing in the step S44, the process returns to the step S3.

The game CPU core 50 processes flowcharts shown in Figures 39 to 43. First, a state of the game CPU flag is determined in a step S201. When it is determined to be in the set state, key assignment request processing is executed in a step S202, and game CPU display processing is executed in a step S203.

The key assignment request processing is executed according to a subroutine

shown in Figure 41. Referring to Figure 41, the key assignment request, the key code indicative of "# key", the function information indicative of "select", the key code indicative of "* key", the function information indicative of "start" and the end code are set to the FIFO register 1 in steps S231 to S236. Successively, in step S237, the interruption request is set to the interruption register 1 shown in Figure 13, and after the completion of the process in the step S237, the process is restored to the hierarchical upper level of the routine. As a result, the key assignment processing (step S38 in Figure 24) is executed by the phone CPU core 28a, and therefore the selection function is assigned to the # key 24e and the start function is assigned to the * key 24d.

The game CPU display processing is executed according to a subroutine shown in Figure 42. Referring to Figure 42, first, the static character data forming the background image data is set to the VRAM 72 in a step S 241, and the palette data of the static character is set to the color palette RAM 68 in a step S242. Successively, it is determined whether or not the screen to be displayed at this time is a screen to output the object image data in a step S243, and if "NO", the process directly proceeds to a step S247. However, if "YES", the process proceeds to the step S247 through the processing of steps S244 to S246. For example, since the object image (dynamic character) is needed to display the game screen shown in Figure 4, the process proceeds to the step S244 at this time. On the other hand, it is appropriate that only the background image is displayed in the player name input screens in Figures 6(A) and 6(B) and therefore, the process proceeds to the step S247.

If displaying the object image, first the dynamic character data is set to the VRAM 74 in the step S244, and the palette data of the dynamic character is set to the color palette RAM 68 in step S245. Furthermore, the attribute of the dynamic character is set to the OAM 76 in the step S246, and in completion of processing of the step S246, the process

proceeds to the step S247.

The settings relating to the output of the background image and the object image are set to the register 70 in the step S247, and it is determined whether the processing for one frame is finished or not in a following step S248. Herein, if "NO", the process returns to the step S241 so as to repeat the above-described processing. If "YES" , the process is restored to the hierarchical upper level of the routine.

The PPU 78 performs the coloring on the static character data according to the palette data of the color palette RAM 68, in accordance with the settings of the register 70, and performs the coloring on the dynamic character data according to the palette data of the color palette RAM 68 as necessary. The color image data is written to the frame memory 88 by the LCD controller 84 and then is output to LCD 18 by the same LCD controller 84. As a result, the color image is displayed on the LCD 18.

The menu screen shown in Figure 3 is displayed on the LCD 18 in the game CPU display processing of the step S203 shown in Figure 39. The menu screen is formed by the static character (text) and the object (cursor).

It is determined whether or not the start key (* key 24d) is operated in a step S204, and it is determined whether or not the select key (# key 24e) is operated in a step S205. Both the determinations are executed referring to the game key register 1. When the select key 24e is operated, the subroutine shown in Figure 42 is processed in a step S207, and then the process returns to the step S204. The cursor on the menu screen is moved by the process of the step S207. When the start key is operated, the selected item is fixed in a step S206. For example, when the start key is pressed in a state where the cursor points to "SUPER MARIO DX", as shown in Figure 3, "game" is fixed as the selected item. When the start key is pressed in a state the cursor points to "GAME OVER", "GAME OVER" is fixed as the selected item.

It is determined whether or not the selected item is "GAME OVER" in a step S208, and it is determined whether or not the selected item is "GAME" in a step S209. If the selected item is "GAME OVER", the processes after a step S211 is executed, and if the selected item is "GAME", the processes after step S214 is executed. If the selected item is neither "GAME" nor "GAME OVER", the corresponding process is executed in a step S210, and then, the process returns to the step S203.

The key assignment clearing is set to the FIFO register 1 in the step S 211, and the interruption request is set to the interruption register 1 in a following step S212. Accordingly, the function assignment table 34a is cleared by the phone CPU core 28a (step S40 in Figure 24). In completion of the processing of the step S212, the game CPU start flag is reset in a step S213 so as to complete the processing.

Referring to Figure 40, game processing, game CPU display processing, and effective sound output processing are executed in steps S214, S215 and S216. In the game processing, the settings of the color palette RAM 68, the register 70, the VRAMs 72 and 74, and the OAM 76 are renewed as necessary. In the game CPU display processing, the game image formed by the screen, the static object and the dynamic character shown in Figure 4 is displayed on the LCD 18. In the effective sound output processing, the effective sound data is output from the SPU 64. In a step S217, it is determined whether or not the start key is operated by referring to the game key register 1, and it is determined whether or not the game is over in a step S218.

When the game is over, "YES" is determined in the step S218, and the process returns to the step S203. Thus, the menu screen shown in Figure 3 is displayed on the LCD 18.

When K6 bit of the game key register 1 is renewed to "1" by an operation of the start key, "YES" is determined in the step S217, and the game is paused in a step S219.

More specifically, the K6 bit is restored to "0" and current game data is stored in the flash memory 54. The value of the K6 bit of the game key register 1 is identified again in a step S220, and long as the value of the K6 bit indicates "0", the process of the step S220 is repeated. Thus the pause state is continued. When the K6 bit of the game key register 1 are renewed to "1" by an operation of the start key, "YES" is determined in the step S220, and the pause state is canceled in a step S221. More specifically, the K6 bit of the game key register 1 is restored from "1" to "0". In completion of the process of the step S221, the process returns to the step S214. As a result, the movement of the game screen and the BGM are resumed.

It is noted that even if it is impossible to return from the pause state due to any reason (e.g. consumption of a battery) after finishing a phone call, it is possible to resume the game where it is interrupted by another operation (e.g. restart) by storing the game data in the step S219.

If a name input stage appears for player name input during the game processing in the step S214, the game CPU core 50 processes a subroutine shown in Figure 43. First, the game CPU display process shown in Figure 42 is executed in a step S251. Thus, the color image date of the message "INPUT YOUR NAME" and the name input frame as shown in Figure 36(B) are written to the frame memory 88, and the name input screen shown in Figure 36(C) is displayed on the LCD 18. However, the cursor is not displayed when the step S251 is completed.

Successively, the FEP transfer request is set to the FIFO register 1 in a step S252, the cursor display position data is set to the FIFO register 1 in a step S253, and the interruption request is set to the interruption register 1 in a step S254. Accordingly, the FEP transfer processing (step S42 in Figure 24) is executed by the phone CPU core 28a, and the cursor is displayed within the name input frame as shown in Figure 36(C).

It is determined whether or not the interruption request is set to the interruption register 2 in a step S255. If "YES", the setting of the FIFO register 2 (FEP result transmission) is detected in a step S256, and the setting of the FIFO register 2 (JIS code) is detected in a step S257. Furthermore, the game CPU display processing is executed in a step S258 so as to display the name within the name display frame on the basis of the detected JIS code. Hence, the color image date of the message, the name display frame and the name, as shown in Figure 38(B), are drawn in the frame memory 88, and the name input screen shown in Figure 38(C) is displayed on the LCD 18. A drawing position of the color image data indicative of the name is determined on the basis of the cursor display position data described above. In completion of the process of the step S258, the interruption register 2 is reset in a step S259, and then the process is restored to the hierarchical upper level of the routine.

As understood from the above description, when the operator operates the * key 24d to interrupt the game being executed (to shift to a pause state), the phone CPU core 28a sets "1" to the KO6 bit of the game key register 3 in response to the key operation (following this, "1" is set to the K6 bit of the game key register1). The phone CPU core 28a also sets "1" to the above-mentioned KO6 bit when there is an incoming phone call during the game. The game CPU core 50 pauses the game in execution, stores the current game data into the flash memory 54 and restores the K6bit to "0" when the K6 bit of the game key register 1 is renewed to "1". When there is an incoming phone call during the game, the tone of the game screen is modified in addition to a shift to the pause state. If the * key 24d is operated in a state in which the game is paused, the phone CPU 28a sets the setting of the K6bit into "1" again. Then, the game CPU core 50 resumes the game.

Thus, when there is an incoming phone call, the game is not ceased, but merely paused. Accordingly, it is possible to resume the game from the middle after the

completion of the communication. In addition, not only when the * key 24d is operated, but also when there is an incoming phone call, "1" is set to the K6bit of the game key register 1. Accordingly, it is possible to pause the game upon reception of the incoming phone call without changing the game pause function originally provided in a conventional game program.

It is noted that in this embodiment the areas 1 and 2 are formed as shown in Figure 16 and the characters indicative of the receiving condition and the remaining amount of the battery are displayed in the area 1 in a manner shown in Figure 17. However, the areas 1 and 2 may be formed as shown in Figure 44, and the characters indicative of the receiving condition and the remaining amount of the battery in the area 1 may be displayed in a manner shown in Figure 45.

Furthermore, although in this embodiment the tone of the game screen is up when the incoming call of the phone is received, the tone of the game screen may be down.

Furthermore, although in this embodiment the start function and the selection function of the game are assigned to the phone keys (# and *), another functions of the game may be assigned to the phone keys.

Although the illustrative embodiments have been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the illustrative embodiments being limited only by the terms of the appended claims.